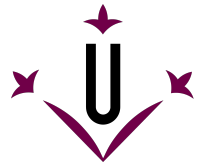

Universitat de Lleida

HONEYPOTS

Carles Mateu

GREiA Research Group
Universitat de Lleida

Outline

Honeypots

Tools

Outline

Honeypot

The same way a real-world honeypot attracts flies, a honeypot attracts hackers to a trap.

Honeypot

A honeypot is a **trap** set to detect, deflect, or counteract malicious attempts.

It is a computational resource designed to be used without authorization and illicitly, it's value lies in being used "wrongly".

Honeypot

Their purpose is:

- As a research tool, to gather knowledge on techniques, tools, intents, targets, etc.
- As an IDS, to detect attacks early.
- As a deflection tool, offering “better” (as in easier) targets to lure attackers away.

Honeypot: goals

- Host connected to a network, either real or virtual host.
- Should look as realistic as possible
- Should look attractive to attackers
- Should be **actively** monitored:
  - To detect it being attacked.
  - To avoid it being used for pivoting.

Honeypots: taxonomy

Honey pots can be classified as:

- High / Low interaction
- Virtual / Physical implementation
- Production / Research oriented

Honeypots: interaction

Low interaction

- Have limited interaction: emulating only services (even operating systems)
- Only simulate partially services: cannot be fully exploited to get complete access
- Responses (and attacks) are limited to the emulated level

High interaction

- Usually more complete (and complex) implementations, involving real services and OS
- No emulation/simulation. Real services are exposed.
- The honeypot can be compromised totally. Allowing attackers full access to the honeypot system, and to use it for further penetration attacks.

Honeypots: interaction

Low interaction

- Have limited interaction: emulating only services (even operating systems)
- Only simulate partially services: cannot be fully exploited to get complete access
- Responses (and attacks) are limited to the emulated level

High interaction

- Usually more complete (and complex) implementations, involving real services and OS
- No emulation/simulation. Real services are exposed.
- The honeypot can be compromised totally. Allowing attackers full access to the honeypot system, and to use it for further penetration attacks.

Honeypots: implementation

Physical

- Real machines deployed as honeypot
- Own IP addresses and services
- High interaction (usually)

Virtual

- Simulated by other machines running the simulation
- Many honeypots on the same server

Virtual doesn't mean virtual machines as VirtualBox

Honeypots: implementation

Physical

- Real machines deployed as honeypot
- Own IP addresses and services
- High interaction (usually)

Virtual

- Simulated by other machines running the simulation
- Many honeypots on the same server

Virtual doesn't mean virtual machines as VirtualBox

Honeypots: implementation

Physical

- Real machines deployed as honeypot
- Own IP addresses and services
- High interaction (usually)

Virtual

- Simulated by other machines running the simulation
- Many honeypots on the same server

Virtual doesn't mean virtual machines as VirtualBox

Honeypots: orientation

Production

- Deployed on real production network.
- Usually low interaction.
- Purposed as deflection/early warning systems.
- Do NOT work against automated attacks.
- Should be easy to deploy/pull offline.

Honeypots: orientation

Research

- Usually high interaction and complex.
- Purposed for discovery of new methods, tools, tactics, etc.
- Gather intelligence to understand motives, organization, behaviour, targeting,...
- Used as forensics and analytics tools, not for deflection.
- Used to capture extensive data.
- Usually deployed by research, government, military, etc.

Honeypots: Pros

- Small data sets, highly valuable.
- Easy to analyze the data.
- Designed to capture **everything** going to them, even unseen tools, techniques, etc.
- Relatively cheap
- Work fine even in complex secure networks
- Collect in-depth information
- Simple (conceptually)

Honeypots: Cons

- Only directed attacks can be captured
- Risky (you are exposing a machine)
- Time consuming: maintaining a high interaction honeypot takes time
- Sometimes can be difficult to realize/analyze what happened
- High interaction honeypots are risky (full fledged machines designed to be attacked and compromised)
- Low interaction honeypot can be detected (sometimes easily)

Legal

There is some dispute on the legality of honeypots:

It seems that you cannot initiate legal actions based on honeypot abuse, as you are luring the attack by creating a target.

Honeynets

- Networks of cooperating honeypots
- Usually with a special firewall (honeywall, f.i.) monitoring all traffic
- Based on the principle that: NO TRAFFIC should go to the honeypots, so all traffic is an attack
- Really good to gather information (not used for deflection)

Client Honeypot

- *Reversed* honeypots, designed to act as clients to internet services (most common is web browsers).
- Detect attacks by compromised or malicious servers
- Useful to detect new exploits and new malware

HoneyMonkey

Research project (Microsoft) of a *reversed* honeypot (honeyclient):

- Network of computers simulating web users using browsers to surf the net.
- Detect the malware injection by malicious or comprimised websites
- Works by comparing a memory image before and after visiting a website
- Useful for detecting 0-day attacks

Outline

Honeyd

- Most known project (although not really maintained)
- Open Source
- Operation:
  1. Monitors access to unused IPs on the network
  2. Detects those access intents and takes over the IP (ARP spoofing)
  3. Creates a virtual honeypot to fool attackers

Project Honeynet

- Honeynet effort targeted towards SPAM
- Crowdsourced effort, anyone can join, to pinpoint SPAM generators

`https://www.projecthoneypot.org/`

Cheapo honeypots

With the advent of SBCs like the Raspberry Pi, it's really cheap to deploy honeypots/honeynets. With projects like:

- `http://mushmush.org/`
- `https://sourceforge.net/projects/honeeepi/`
- `https://github.com/mattymcfatty/HoneyPi`

You can easily build a cheapo network of honeypots.

Awesome list

As always, there's an awesome list for honeypots:

`https://github.com/paralax/awesome-honeypots`